

# Nicholas-Applegate Capital Management

## A NeXT Success Story

*In three years, Nicholas-Applegate was able to reengineer its business systems, enabling the company to manage its business growth from \$4 to \$14 billion in assets.*

In the highly volatile financial management marketplace, companies are only as good as their information. The industry is largely data driven, with traders, investment advisors, portfolio managers, and researchers relying on independent data sources from multiple vendors. To gain a competitive advantage, a successful company must be able to intelligently combine real-time data with trade history in a meaningful way. It must have the ability to quickly turn data into information—which translates into knowledge.

Using object technology from NeXT Computer, Inc., Nicholas-Applegate was able to develop an investment and trading environment that was flexible and able to expand as the company grew. Under the direction of Duncan Wilcox, Director of Investment Technology, Nicholas-Applegate developed NEXTSTEP™ applications that combine real-time data feeds and relational data with business logic to provide valuable information in a heterogeneous client/server environment.

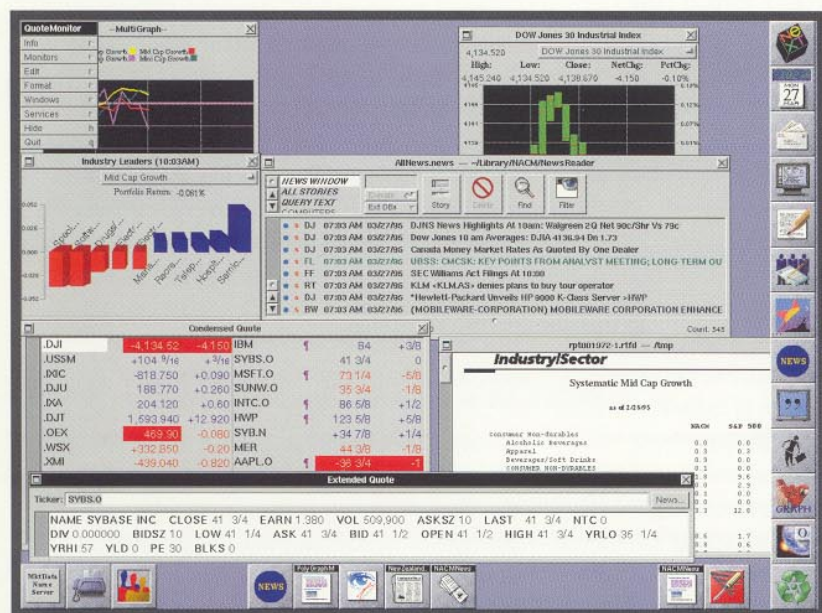
### Company Challenges

Nicholas-Applegate needed a system that was:

- scalable so that it could easily meet the company's aggressive growth plan
- flexible and able to adapt to a changing client/server environment

After evaluating several 4GL application development environments, Nicholas-Applegate chose object-oriented development tools from NeXT. They chose NEXTSTEP Developer because it would allow the development team to build applications in half the time it would have taken with other development tools. NeXT's Enterprise Objects™ Framework enabled Nicholas-Applegate to combine data from multiple data sources with business logic into reusable objects. And with NeXT's Portable Distributed Objects™ —PDO™—their applications could scale by distributing objects across heterogeneous clients and servers running SunOS, Solaris, HP-UX, and Digital UNIX operating systems.

The development team's goal was to demonstrate that it could develop trading applications that combined real-time data feeds with existing data sources to provide value-added information to the end user and then deploy those applications across multiple hardware platforms and operating systems. Traders had been gathering quotes and pricing their portfolios using Excel spreadsheets. Wilcox knew that with NEXTSTEP they could do much more.



Nicholas-Applegate Capital Management of San Diego, California, knew it needed to overhaul its information systems if it was to successfully grow its business and continue to compete with larger financial firms, especially those on Wall Street. Nicholas-Applegate's problem wasn't that it didn't have enough data. Its problem was that it was data rich, but information poor.

In three years, Nicholas-Applegate reengineered its business systems, enabling the company to manage its business growth from \$4 to \$14 billion in assets.





Using NeXT's tools, including NEXTSTEP Developer, Enterprise Objects Framework, and PDO, the development team built its first trading application. This application included objects that price portfolios in real time and allow traders and analysts to view portfolios and holdings simultaneously. Traders have the ability to further analyze a particular security against historical data, allowing them to compare up-to-the-second information about a particular stock with its past performance in account holdings.

*"Because PDO provides transparent distribution, local objects can be easily distributed. It simplifies our development effort because we can run objects where they make the most sense and painlessly relocate objects as needed." — Duncan Wilcox, Nicholas-Applegate*

#### Portable Distributed Objects

Due to the nature of its business, Nicholas-Applegate is a compute-intensive environment. Therefore, it was imperative that the team develop a system that used the capabilities of its hardware to its best advantage. Portable Distributed Objects provides a heterogeneous environment based on objects by extending NEXTSTEP's dynamic object model to non-NEXTSTEP operating systems. Taking advantage of the flexibility of PDO, Nicholas-Applegate was able to use its network in ways that made sense both financially and technologically. PDO allowed Nicholas-Applegate's development team to make hardware decisions based on system needs rather than on system compatibility.

The team first deployed its applications on NEXTSTEP for Intel. As the system expanded, Nicholas-Applegate needed to incorporate powerful servers to distribute objects with compute-intensive processing logic. PDO offered the multiplatform capabilities that Nicholas-Applegate needed, enabling it to create a heterogeneous client/server environment that incorporated HP 9000 and Sun® SPARC servers.

Keeping in mind the anticipated growth of the company, developers also needed a system that could expand and adapt as more users were added to the network. Since initial deployment, the number of system users has tripled. Nicholas-Applegate's development team has kept up with this increased demand by adding servers and desktops as needed.

"Because PDO provides transparent distribution, local objects can be easily distributed," said Wilcox. "It simplifies our development effort because we can run objects where they make the most sense and painlessly relocate objects as needed. PDO allows us to plug in whatever we need as the system grows."

"Without the benefits of PDO, we would have had to get down to lower computing levels and do some serious coding," said Wilcox. "PDO allows us to locate objects where they make the most sense—if the business logic involves long calculations, it can live on the server; if it is a real-time feed, the object can live on the desktop. This allows us to concentrate on strategic architecture decisions and significantly reduces our training and overhead costs."

Similarly, PDO allowed the Nicholas-Applegate development team to integrate data services from other vendors such as Bridge, from Bridge Data Systems, and VISION, by Innovative Systems Techniques, Inc., as well as access external data pipes like IDC (Interactive Data Corporation). By using PDO to objectify connections to other information systems running on other operating systems, such as Solaris, developers are able to treat what would otherwise be foreign and inaccessible data as integral parts of the NEXTSTEP object universe. UNIX® services are encapsulated within a NEXTSTEP object interface, allowing developers to build easy-to-use interfaces even onto legacy information systems. The result: end users maintain a consistent NEXTSTEP user interface while developers gain a unified, objectified entry point to external data.

The development team at Nicholas-Applegate has also used the unique capabilities of NeXT's dynamic objects throughout its custom applications. For example, early on in its development of core financial applications, the team developed a "Ticker Picker" object that is loadable on demand into any NEXTSTEP application. Nicholas-Applegate's Ticker Picker provides a consistent interface for users to find ticker symbols for any traded security. The Ticker Picker is a dynamic object. Therefore, its capabilities can be extended or refined by developers at any time—without the need to recompile the applications that use its services.

*"Using the Enterprise Objects Framework, we no longer have to recode our applications every time a change is made to the database architecture, as we would have had to with any other database development tool." — Duncan Wilcox, Nicholas-Applegate*

#### Enterprise Objects Framework

Not only is Nicholas-Applegate an extremely compute-intensive environment, it's also an extremely data-intensive environment, with data retrieved from relational databases, real-time data feeds, and external commercial databases. Using NeXT's Enterprise Objects Framework, a tool that allows developers to combine persistent data from multiple data sources with business logic to create reusable objects, Nicholas-Applegate's developers are able to forge enterprise data into something far more valuable—business objects.

The Enterprise Objects Framework allowed Nicholas-Applegate to incorporate the full power of its existing data sources within its overall object-oriented development strategy. Now data can be treated at a much higher conceptual level—as objects. The Enterprise Objects Framework insulates objects from the details of how data is stored in the underlying database, enabling changes to the database schema

to be made without having to make changes to the objects themselves, and vice versa.

"Using the Enterprise Objects Framework, we no longer have to recode our applications every time a change is made to the database architecture, as we would have had to with any other database development tool," according to Wilcox.

When combined with PDO, the Enterprise Objects Framework allows developers even more flexibility in determining the architecture of the system. Using the Enterprise Objects Framework and PDO, persistent objects can be distributed across a network of clients and servers.

"When we need to perform calculations against 25,000 rows of data, we don't want that data coming across the network," said Wilcox. "We only want the result of the computation reaching the desktop application. The Enterprise Objects Framework and PDO make that possible."

The combination of the Enterprise Objects Framework and PDO allows the system to accommodate a growing number of users, as well as massive amounts of data and additional complex business logic.

In addition to the NeXT-supplied SYBASE® adaptor for the Enterprise Objects Framework, Nicholas-Applegate developed a number of custom Enterprise Objects Framework data source adaptors that enabled them to bring real-time data and historical pricing information into the Enterprise Objects Framework object model. For the first time, Nicholas-Applegate's development team was able to develop custom applications that allowed the end user to look at the full universe of financial data, rather than just a particular vendor's piece. The Enterprise Objects Framework provided the object foundation that made this combination of information possible.





For additional information,  
call 1-800-TRY-NeXT

NeXT Computer, Inc.  
900 Chesapeake Drive  
Redwood City, CA 94063

## Looking Forward

Now that Nicholas-Applegate understands its investment in object technology—and the vast capabilities it provides—the company's development team has begun to develop its own suite of in-house financial objects. These objects will take full advantage of the NEXTSTEP/OpenStep™ environment to encapsulate the knowledge of Nicholas-Applegate's financial expertise in a way that best supports the unique needs of the firm's money managers.

What started as an intensive and targeted application development strategy is now shifting to encompass a company-wide deployment strategy. The development team is focusing on building an information infrastructure throughout Nicholas-Applegate that will provide custom applications to all end users, including departments outside financial services, such as human resources and marketing.

Nicholas-Applegate is also looking forward to OpenStep, which is NeXT's strategy to provide an object-oriented application deployment standard. This strategy includes deployment on operating system environments such as Solaris, Digital UNIX, Windows NT, and Windows 95.

"NEXTSTEP is the foundation of our information systems and will continue to grow and expand with our business," says Wilcox. "We have realized the benefits of, and will continue to develop applications, using NEXTSTEP. Without NeXT's technology, we could not have built these applications in such a short timeframe and deployed them with such success. We owe much of our company's increased profits to the fact that our systems have been able to keep pace with our exploding growth—we couldn't have done it without NeXT."